

УДК 347.440:004.42

Фасій Богдан Володимирович

кандидат юридичних наук, доцент кафедри цивільного права,
заступник декана факультету цивільної та господарської юстиції
Національного університету «Одеська юридична академія»

AGILE, WATERFALL ТА OUT STAFF ДОГОВІР АБО ДОГОВОРИ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Постановка проблеми. Прискорення та розвиток науково-технічного прогресу, інформаційних технологій, комп’ютерної техніки та комп’ютерних мереж в Україні засноване на впровадженні у виробництво гнучких автоматизованих систем, мікропроцесорних засобів і пристройів програмного управління, роботів і обробних центрів, поставило перед сучасною науковою надважливе завдання – виховати та підготувати підростаюче покоління науковців та юристів, здатних активно включитися в якісно новий етап розвитку сучасного суспільства, пов’язаний з інформатизацією. У цій сфері законодавство не здатне стрімко впроваджувати норми і не встигає реагувати на різкі зміни у розвитку суспільних відносин. Це є нормальним явищем не тільки для України, але й для інших європейських країн, навіть найстабільніших. Звичайно, зі спливом часу врегулювання даних суспільних відносин, знайде своє відображення у праві.

Метою статті є дослідження окремих проблем таких договорів на розробку програмного забезпечення як agile, waterfall та out staff договір.

Виклад основного матеріалу. Нині для врегулювання відносин між компанією та підрядником (розробником) програмного забезпечення укладають договори у письмової формі, якщо раптом що – «Дивись підпункт 3, пункту 2 розділу 4!».

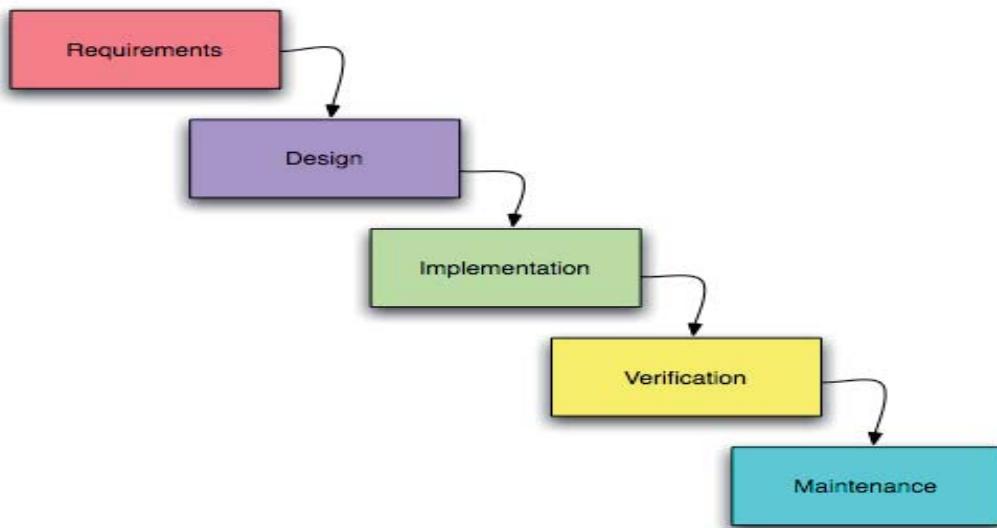
З одного боку, максимально детальні договори, що передбачають всі можливі ситуації, які можуть трапитися під час розробки та після неї, це добре, але ж чи є сенс переписувати цивільний, трудовий, кримінальний, податковий кодекс у договір? Ще в максимально детальних договорах іноді забивають про істотні умови. Хоча з іншого – без договору або з «договором на серветці» у випадку конфлікту інтересів у суді буде не так просто захистити свої права та інтереси.

Звісно, це питання, не таке проблематичне як вибір мови програмування для програміста-початківця, але все ж серйозне.

Waterfall договір (з англ. Waterfall іноді перевідають як модель «Водоспад» або «Каскадна модель») – договір на розробку програмного забезпечення, в якому істотні умови послідовно врегульовують проходження фази аналізу вимог, проектування, реалізації, тестування, передання замовнику, інтеграції та підтримки. Як джерело, часто вказують статтю, опубліковану У.У. Ройсом у 1970 році; при тому, що сам автор використовував договір ітеративної моделі розробки [1].

Дотримуючись каскадної моделі, розробник за waterfall договором переходить від однієї стадії до іншої строго послідовно. Спочатку повністю завершується етап «визначення вимог», у результаті чого виходить список вимог до програмного забезпечення. Після того як вимоги повністю визначені, відбувається перехід до проектування, у ході якого створюються документи, що детально описують для програмістів спосіб і план реалізації зазначених вимог. Після того, як проектування повністю виконане, програмістами за договором виконується реалізація отриманого проекту. На наступній стадії процесу відбувається інтеграція окремих компонентів, що розробляються різними командами програмістів. Після того, як реалізація і інтеграція завершенні, проводиться тестування і налагодження продукту; на цій стадії усуваються всі недоліки, що з’явилися на попередніх стадіях розробки. Після цього програмний продукт передається замовнику, впроваджується і забезпечується його підтримка – внесення нової функціональності і усунення помилок.

Перехід від однієї фази до іншої відбувається тільки після повного і успішного завершення попередньої



Таким чином, каскадна модель має на увазі, що перехід від однієї фази розробки до іншої відбувається тільки після повного і успішного завершення попередньої фази, і що переходів назад, вперед або перекриття фаз – не відбувається.

Проте, існують модифіковані каскадні моделі (включаючи модель самого У.У. Ройса), що мають невеликі або навіть значні варіації описаного процесу.

У таких waterfall договорах чітко вказується технічне завдання, яке, зазвичай, узгоджується сторонами у додатках та відразу підписується. Бувають кейси, коли замовник підписує таке технічне завдання, при цьому не читаючи його, а потім, коли результат не відповідає очікуванням, виявляється, що в самому технічному завданні також вказано зовсім не те, що хотів би бачити замовник. У таких випадках замовнику, який не розуміється у технічних нюансах і для якого технічне завдання виглядає наче ієрогліфи, перед укладенням договору, краще проконсультуватись з іншим незалежним розробником перш ніж затверджувати технічне завдання на предмет того, чи відповідає воно його очікуванням. Якраз відповідність технічному завданню в судовій практиці часто стає предметом експертизи і вирішальним у спорах про неякісне надання послуг стосовно розробки програмного забезпечення.

Наступним договором на розробку програмного забезпечення є agile договір. В agile договорі закріплюються основні лейтмотиви, ідеї та концепції гнучкої розробки – а саме, постійна співпраця між замовником та підрядником (розробником).

Сторони agile договору скоріше добре партнери, ніж замовник та підрядник (розробник). В agile договорі у будь-якому випадку мають бути істотні умови, характерні для договорів розробки софту.

Всі детальні умови комунікації між сторонами можна розмістити у відповідному розділі – «Взаємодія Сторін». Калькуляцію вартості конкретних User Story можна також розписати, так само як і спринти – етапи в рамках загального строку надання послуг.

Такі види договорів набирають все більшої популярності серед підрядників (розробників) та замовників, оскільки досить часто процес розробки є довготривалим і умови ринку, під який створюється продукт змінюються. Саме адаптивність до можливих змін, постійна співпраця і націленість обох сторін на успішний реліз продукту є основною перевагою agile договорів.

Основу agile договорів становивить гнучка методологія розробки (англ. Agile software development, agile-методи) – серія підходів до розробки програмного забезпечення, орієнтованих на використання інтеративної розробки, динамічне формування вимог і забезпечення їх реалізації в результаті постійної взаємодії всередині самоорганізованих робочих груп, що складаються з фахівців різного профілю [2]. Існує кілька методик, що відносяться до класу гнучких методологій розробки, зокрема екстремальне програмування, DSDM, Scrum, FDD.

Agile-методи роблять акцент на безпосередньому спілкування віч-на-віч. Більшість agile-команд розташовані в одному офісі, іноді так званому англ. bullpen. Як мінімум, вони включають і «замовників» (англ. Product owner – замовник або його повноважний представник, який визначає вимоги до продукту; цю роль може виконувати менеджер проекту, бізнес-аналітик або клієнт). Офіс може також включати тестувальників, дизайнерів інтерфейсу, технічних письменників і менеджерів.

Основною метрикою agile-методів є робочий продукт. Віддаючи перевагу безпосередньому спілкуванню, agile-методи зменшують обсяг письмової документації в порівнянні з іншими методами.

Більшість гнучких методологій націлені на мінімізацію ризиків шляхом зведення розробки до серії коротких циклів, йменованих ітераціями, які зазвичай тривають два-три тижні. Кожна ітерація сама по собі виглядає як програмний проект в мініатюрі і включає всі завдання, необхідні для видачі міні-приросту по функціональності: планування, аналіз вимог, проектування, програмування, тестування і документування. Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі, що гнучкий програмний проект готовий до випуску в кінці кожної ітерації. Після закінчення кожної ітерації команда виконує переоцінку пріоритетів розробки [3].

У лютому 2001 року в штаті Юта США був випущений «Маніфест гнучкої методології розробки програмного забезпечення» (англ. Agile Manifesto). Він був альтернативою розробки програмного забезпечення, такого як «waterfall метод», який був золотим стандартом розробки в той час. Даний маніфест був схвалений і підписаний представниками методологій: екстремального програмування, Crystal Clear, DSDM, Feature driven development, Scrum, Adaptive software development, Pragmatic Programming. Гнучка методологія розробки використовувалася багатьма компаніями і до прийняття маніфесту, однак входження Agile-розробки в суспільство відбулося саме після цієї події.

Agile Manifesto роз'яснює 12 принципів, якими є:

1. задоволення клієнта за рахунок ранньої і безперебійної поставки програмного забезпечення;
2. заохочення змін вимог навіть в кінці розробки (це може підвищити конкурентоспроможність отриманого продукту);
3. постійна поставка робочого програмного забезпечення (кожного місяця або тижня або ще частіше);
4. тісне, щоденне спілкування замовника з розробниками протягом усього проекту;
5. проектом займаються мотивовані особистості, які забезпеченні необхідними умовами роботи, підтримкою і довірою;
6. рекомендований метод передачі інформації – особиста розмова (віч-на-віч);
7. працююче програмне забезпечення – кращий вимірювач прогресу;
8. спонсори, розробники і користувачі повинні мати можливість підтримувати постійний темп на невизначений термін;

9. постійна увага поліпшенню технічної майстерності і зручному дизайну;
10. простота – мистецтво не робити зайвої роботи;
11. кращі технічні вимоги, дизайн та архітектура виходять у самоорганізованих командах;
12. постійна адаптація до обставин, що змінюються. Команда повинна систематично аналізувати можливі способи поліпшення ефективності і відповідно коригувати стиль своєї роботи [4].

Ще одним варіантом урегулювання IT-відносин може бути *out staff* договір. У цьому випадку замовник отримує у своє розпорядження певні години конкретних спеціалістів зі професійними навичками, які описані в договорі. Такі спеціалісти надають послуги в межах заявлених навичків. При цьому, роботу таких спеціалістів може координувати менеджер, години якого також продаються в рамках *out staff* договору. Такий варіант підходить для досвідчених замовників, або тих, хто має розуміння у сфері розробки софту, і відповідно, зможе отримати користь саме від такого алгоритму роботи.

У таких IT-договорах фіксуються кількість годин конкретних спеціалістів, вартість таких годин, у залежності від спеціалізації та навиків осіб, що залучені.

Пояснюючи складне простими словами, можна сказати, що це один з численних видів управління персоналом. Суть *out staff* в тому, що спеціалісти виводяться за штат компанії або організації, офіційно числяться в іншій юридичній особі або в стартапі, але продовжують працювати на колишньому робочому місці. Фактично ці співробітники беруться в оренду на термін, визначений у договорі. Інколи це кілька місяців, а іноді кілька років – все залежить від компанії-наймача.

Вперше подібна практика була застосована в Сполучених Штатах Америки в 70-х роках минулого століття, у період одного з серйозних економічних криз. До України така форма кадрового менеджменту прийшла в 1998 році, коли багато компаній розорювалися, а решта намагалися всіляко мінімізувати штат співробітників. Нині послуги *out staff* персоналу надають практично всі великі кадрові агентства, і з кожним роком популярність цієї форми найму тільки зростає.

Дуже часто *out staff* за співзвучністю плутають з outsourcing, проте це абсолютно різні поняття. Outsourcing являє собою передачу компанією тих чи інших непрофільних для неї функцій стороннім організаціям за договором (наприклад, IT-обслуговування). Як приклад, наведемо тактику

Apple Inc™, однією з найпопулярніших і найбагатших компаній в світі. Ця компанія – лідер в частині грамотного використання персоналу та передачі функцій стороннім організаціям. Як відомо, Apple не виготовляє телефони та планшети самостійно. Всі компоненти і фінальна збірка виробляються в Китаї та Південній Кореї, що представляє собою типовий outsourcing. А для розробки передових моделей або нових пристрій беруться в оренду дизайнери і вузькі фахівці в тій чи іншій області. Зокрема, до роботи над iWatch (годинник від «яблучної» компанії) було залучено близько сотні тимчасових фахівців з провідних годинних фірм Швейцарії [5].

Out staff договір доцільний в наступних випадках:

- рішення разових завдань;
- виконання сезонних робіт;
- тимчасове заличення людей для виконання великого обсягу роботи в стислі терміни;
- тимчасова заміна персоналу на період відсутності (щорічна відпустка, декретна відпустка, тривалий лікарняний і т.п.);
- відсутність певних необхідних категорій працівників;
- відсутність часу на пошук фахівців потрібної кваліфікації [6].

Out staff як спосіб управління персоналом має безліч переваг. Зокрема, в умовах фінансової нестабільності це дозволяє на законних підставах

мінімізувати податки, скоротити витрати на персонал і позбутися від мало завантажених (або зовсім надлишкових) співробітників. Якраз тому out staff з кожним роком користується все більшою популярністю, причому не тільки в великих містах, а й в невеликих, провінційних.

Для співробітника, що працює так, єдиним мінусом може бути те, що він не може користуватися додатковими соціальними послугами компанії-замовника, так як складається в офіційних трудових відносинах з компанією-out staff.

Висновки. Звісно, весь цей поділ договорів на розробку софту на waterfall договір, agile договір та out staff договір є досить умовним. Ключові відмінності були вказані вище і в залежності від конкретного кейсу та суті правовідносин можна обрати потрібний формат договору. А ще – можна розробити певний mixed agreement, який буде включати елементи всіх трьох договорів. Про те слід пам'ятати, що при укладені цих договорів необхідно керуватися нормами Цивільного кодексу України (про це будуть наступні дослідження).

У будь-якому разі для замовника важливо отримати кінцевий результат та всі виключні майнові авторські права на нього. Все це замовнику хочеться отримати за розумну плату та у визначений строк. Для розробника важливо зрозуміло і чітко врегулювати свої обов'язки та виконавши їх, отримати оплату, а також не вказати в договорі того, що гарантувати виконавець не може.

ЛІТЕРАТУРА

1. Каскадная модель. – Режим доступу: https://ru.wikipedia.org/wiki/Каскадная_модель
2. What is Agile? – Режим доступу: <https://www.agilealliance.org/agile101>
3. Гибкая методология разработки. – Режим доступу: https://ru.wikipedia.org/wiki/Гибкая_методология_разработки
4. 12 Principles Behind the Agile Manifesto. – Режим доступу: <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto>
5. Аутстаффинг – что это? Разъяснение термина, способы применения, достоинства и недостатки. – Режим доступу: <http://workstaff.ru/autstaffing-cto-eto-razysnenie-termina-sposoby-primeneniya-dostoinstva-i-nedostatki-pri-ispolzovanii>
6. Аутстаффинг – это актуально. – Режим доступу: <http://workstaff.ru/autstaffing-eto-aktualno>

Фасій Богдан Володимирович

AGILE, WATERFALL ТА OUT STAFF ДОГОВІР АБО ДОГОВОРИ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті досліджуються окрім проблеми IT-договорів на розробку програмного забезпечення. Досліджується порівняльна характеристика agile, waterfall та out staff договору. Встановлено, що в залежності від конкретного кейсу та суті правовідносин можна обрати потрібний формат договору. Також є можливість розробити певний mixed agreement, який буде включати елементи всіх трьох договорів.

Ключові слова: IT-право, IT-відносини, IT-договір, програмне забезпечення, методологія розробки.

Фасій Богдан Владимирович

AGILE, WATERFALL И OUT STAFF ДОГОВОР АБО ДОГОВОРИ НА РАЗРАБОТКУ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В статье исследуются отдельные проблемы ИТ-договоров на разработку программного обеспечения. Исследуется сравнительная характеристика agile, waterfall и out staff договора. Установлено, что в зависимости

от конкретного кейса и сущности правоотношений можно выбрать нужный формат договора. Также есть возможность разработать определенный mixed agreement, который будет включать элементы всех трех договоров.

Ключевые слова: ИТ-право, ИТ-отношения, ИТ-договор, программное обеспечение, методология разработки.

Fasii Bohdan Volodymyrovych

AGILE, WATERFALL AND OUT STAFF AGREEMENTS OR CONTRACTS FOR THE SOFTWARE DEVELOPMENT

The article deals with the specific problems of IT-contracts for software development. The comparative characteristics of the agile, the waterfall and the out staff contracts are being studied. It is established that the right format of the contract can be chosen depending on the particular case and the essence of the legal relationship. It is also possible to develop a specific mixed agreement that will include elements of all three agreements.

Key words: IT-law, IT-relations, IT-contract, software, development methodology